UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/574,165	05/18/2000	Gary S. Burd	MS 144239.1/40062.49US01	6904

22801 7590 03/30/2004

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
NARAYANASWAMY, SINDYA	
ART UNIT	**PAPER NUMBER**
2174	18

DATE MAILED: 03/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/574,165	BURD ET AL.	
	Examiner	Art Unit	
	Sindya Narayanaswamy	2174	

-- The MAILING DATE of this communication appears on th cover sheet with th correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 February 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *14, 15*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 - 37 are presented for examination.

2. Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn. As a result of the discussions during the 2/10/04 interview, Examiner was convinced of the invalidity of the previous final action.

Claim Rejections - 35 USC § 102(e)

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(e) that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1-37 are rejected under 35 U.S.C. 102(e) as being anticipated by Emmelmann, US 2003/0074634.

5. As per claims 1, 16 and 17 Emmelmann teaches the method, signal and computer program storage medium for performing server-side processing of postback input received from a client and associated with a client-side user interface element (pressing of a button on browser display), the method comprising: examining the postback input to determine an identifier of a target server-side control object; identifying the target server-side control object based on the identifier of the target server-side control object; passing the postback input to the target server-side control object; and processing the postback input passed to the target server-side control

object (*server side web application, processes information which is sent to server as a request and then passed to the corresponding interactive server side component*) (page 2, column 1, paragraph 5; page 2, column 2, paragraph 2).

6. As per claim 2, Emmelmann teaches the method wherein the processing operation comprises changing a property of the target server-side control object; and further comprising: generating authoring language data from the target server-side control object based on the property to define the client-side user interface element for transmission to the client (*components can be added to the server side application in an authoring language such as HTML or XML*) (page 3, column 2, paragraphs 3, 4, 9).

7. As per claim 3, Emmelmann teaches the method wherein the processing operation comprises raising a server-side event from the target server-side control object; and further comprising: generating authoring language data from the target server-side control object based on the property to define the client-side user interface element for transmission to the client (page 4, column 1, paragraphs 6-7).

8. As per claim 4, Emmelmann teaches the method creating a plurality of server-side control objects in a server-side control object hierarchy prior to the operation of processing the post-back input; and terminating the plurality of server side control objects, after the operation of generating authoring language data (*enclosed components create a hierarchy*) (page 4, column 1, paragraph 8; page 4, column 2, paragraphs 3-4).

9. As per claim 5, Emmelmann teaches the method comprising: searching for the target server-side control object in a server-side control hierarchy based on the identifier; creating the target server-side control object in the server-side control hierarchy, if the target server-side control object is not found by the searching operation; and terminating (*garbage collection*) the server-side control hierarchy, after the operation of generating authoring language data (page 3, col. 2, paragraph 7, page 5, col. 2, paragraphs 4-5, page 7, col. 2, paragraph 3).

10. As per claim 6, Emmelmann teaches the method wherein the identifier has a hierarchical identifier structure indicating a plurality of levels in a server-side control object hierarchy including a plurality of member server-side control objects, and the operation of identifying the target server-side control object comprises: extracting a node level identifier from the identifier; passing the node level identifier to a member server-side control object corresponding to the node level identifier; identifying the member server-side control object as the target server-side control object, if the node level identifier identifies a leaf node of the identifier; extracting a next node level identifier from the identifier of the target server-side control object, if the node level identifier does not identify a leaf node of the identifier, wherein the next node level identifier identifies a child server-side control object of the member server-side control object; and performing recursively the passing and identifying operations and the operation of extracting a next node level identifier using the next node level identifier as the node level identifier and the child server-side control object as the member server-side control object, if the node level

identifier does not identify a leaf node of the identifier (*identification of children and parent nodes*) (*contains modules and sub-modules*) (page 5, paragraphs 4-5, Fig. 7, page 6, col. 1, paragraph 9-col. 2, paragraph 1).

11. As per claim 7, Emmelmann teaches the method wherein the operation of processing the postback input comprises: storing a postback data value as a property the target server-side control object (*method of storing user input/data*) (page 5, col. 2, paragraph 1).

12. As per claim 8, Emmelmann teaches the method wherein the target server-side control object initially stores an old data value as a property, and the operation of storing a postback data value comprises: associating the postback data value with the property; indicating a data change associated with the target server-side control object, if the postback data value passed to the target server-side control object is different than the old data value of target server-side control object; and replacing the old data value with the postback data value in the target server-side control object (*persistent components*) (page 9, col. 1, paragraph 2).

13. As per claim 9, Emmelmann teaches the method comprising: raising a server-side data change event after the operation of replacing of the old data value, if a data change indicated (page 9, col. 1, paragraphs 2-3).

14. As per claim 10, Emmelmann teaches the method wherein the target server-side control object is one of a plurality of member server-side control objects in a server-side control object

hierarchy, and the operation of storing a postback data value comprises: storing postback data values for all of the member server-side control objects in the server side control object hierarchy; and raising at least one server-side data change event after the operation of storing a postback data value for all member server-side control objects, if at least one data change is indicated (*component classes 1-n*) (Fig. 7, page 5, col. 2, paragraph 9-page 6, col. 1, paragraph 1).

15. As per claim 11, Emmelmann teaches the method comprising receiving the server-side data change event from the target server-side control object; and invoking a function of a non-user-interface server component, based on the server side data change event (*internal counter*) (page 7, col. 1, paragraphs 8-9).

16. As per claim 12, Emmelmann teaches the method wherein the operation of processing the postback input comprises: processing a postback event using the target server-side control object (*component (server-side) processor*) (page 6, col. 1, paragraphs 2-4).

17. As per claims 13, 14 and 15, Emmelmann teaches the method wherein the operation of processing a postback event comprises: extracting from the postback input a postback event argument associated with the identifier; passing the postback event argument associated with the identifier to the target serverside control object; processing the postback event argument using the target server-side control object, the operation of processing a postback input further comprises: raising a server-side event from the target server-side control object, responsive to the

operation of processing the postback event argument; receiving the server-side event from the target server-side control object; and invoking a function of a non-user-interface server component, based on the server-side (page 2, column 1, paragraph 5; page 2, column 2, paragraph 2).

18. As per claim 18, it is rejected on the same basis as claims 1 and 2.

19. As per claims 19, 21 and 22, they are similar in scope to claim 1 and are rejected on the same basis as claim 1.

20. As per claim 20, it is rejected on the same basis as claim 5.

21. As per claim 23, Emmelmann teaches a computer system comprising a plurality of server-side control objects in a server-side control object hierarchy on a server, the server side control object hierarchy including a target server side control object associated with a client-side user interface element on a client, such that input data received by the server from the client-side user interface element is passed within the server to the target server-side control object in the server-side control object hierarchy; the plurality of server-side control objects generating authoring language data to define a web page for display on the client (*enclosed components create a hierarchy*) (page 4, column 1, paragraph 8; page 4, column 2, paragraphs 3-4; Fig. 1-3 (*samples of web page display generated for client*)).

22. As per claim 24, Emmelmann teaches a computer program product embodied in a computer readable medium for executing a computer process, the computer process comprising: generating authoring language data from a plurality of server-side control objects at a server to define a page for display on a client, the authoring language data including a script that is tagged to be executed by the server to process input data received from the client (*display method, HTML code (authoring language) is generated*) (page 6, col. 2, paragraph 8-page7, col. 1, paragraph 4).

23. As per claim 25, Emmelmann teaches the computer program product wherein each server-side control object corresponds to a client-side user interface element (*every component page has an associated client-side browser page that can be generated using the component classes*) (page 4, col. 1, paragraph 5).

24. As per claim 26, Emmelmann teaches a method comprising: generating authoring language data from a plurality of server-side control objects at a server to define a web page for display on a client (URL), the authoring language (HTML) data including a script that is tagged to be executed by the server to process input data received form the client (page 4, column 1, paragraph 8; page 4, column 2, paragraphs 3-4; Fig. 1-3 (*samples of web page display generated for client*)).

25. As per claim 27, it is similar in scope to claim 23 and is rejected on the same basis.

26. As per claim 28, Emmelmann teaches the method wherein the processing operation comprises: setting a property value of the identified server-side control object based on the input data (*Values entered for each field (ie Guest_Name) and subsequently edited or set*) (Fig. 4; page 10, col. 2, paragraph 5).

27. As per claim 29, Emmelmann teaches the method wherein the processing operation comprises: raising an event in the identified server-side control object based on the input data (page 6, col. 1, paragraphs 8-10) .

28. As per claim 30, Emmelmann teaches the method of claim 27 further comprising: generating authoring language data from a plurality of server-side control objects at a server to define a page for display on the client, the page including the individual client-side user interface element (*client-side user interface generated (Name, Size, etc.)*) (Fig. 24, component editor) (page 4, column 1, paragraph 8; page 4, column 2, paragraphs 3-4; Fig. 1-3).

29. As per claim 31, Emmelmann teaches the method of claim 27, further comprising generating authoring language data from a plurality of server-side control objects at a server to define a page for display on the client, the authoring language data including a script that is tagged to be executed by the server to process input data received from the client (page 4, column 1, paragraph 8; page 4, column 2, paragraphs 3-4; Fig. 1-3 (*samples of web page display generated for client*)).

30. As per claims 32-36, they are similar in scope to claims 27-31 and are rejected on the same basis.

31. As per claim 37, it is similar in scope to claim 27 and is rejected on the same basis.

32. Applicant's arguments with respect to claim 1-37 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

33. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

a. 5,517,655- method of managing/monitoring events in object oriented system

34. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sindya Narayanaswamy whose telephone number is (703) 305-8473. The examiner can normally be reached on 8 am to 5 pm, first Fridays off. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on (703) 308-0640. The fax phone number for the organization where this application or proceeding is assigned is (703) 305-5404. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-9000.

Kristine Kincaid
KRISTINE KINCAID
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

March 17, 2004

Sindya Narayanaswamy